

5

information that describes the different types of data stored in the file. In embodiments, the extension section may contain information related to all of the data in the file or to a subset of data in the file. The extension section may describe a type of object, a property of an object, a value, or any other type of data stored in the self-describing file. For example, the extension section **102** may contain information describing object **106A**, any of the properties **108A-108C**, or any other type of data. The extension section **102** of the self-describing file **100** contains information that allows an application to correctly preserve content stored in the self-describing file **100** that is unknown to the application accessing the file. As such, the extension section **102** provides applications with information that may be used to update and/or maintain data in the self-describing file **100** even if the application does not natively support the capability or property that the data is related to. In embodiments, the extension section may contain XML, binary, HTML, source code, or information in any other form that an application and/or processor executing the application is capable of interpreting.

Although illustrated as part of the self-describing file **100** in FIG. 1, in alternate embodiments the extension section **102** may exist external to the self-describing file **100**. For example, the extension section **102** may be a separate file that is referenced by the self-describing file **100**. In other embodiments, the extension section may be stored separately from the self-describing file **100** but may be otherwise accessible to the application accessing the self-describing file **100** such as in a directory which the application has access to or on a network accessible by the application, such as, but not limited to, the Internet.

In embodiments, the extension section **102** may contain one or more extensions elements, such as Extension **1 104A**, Extension **2 104B**, and Extension **N 104C**. Although the embodiment illustrated in FIG. 1 provides three extension elements, one of skill in the art will appreciate that the extension section **102** may contain any number of extensions. In embodiments, the extensions are used to define objects, properties, or other types of data stored in the self-describing file **100** which may otherwise be unknown to or not natively supported by an application accessing the self-describing file **100**. For example, the extensions may define a property that the application may not support. In other embodiments, the extensions may define a relationship between properties that the application may support, but may not be aware of the relationship between the properties. For example, referring again to an exemplary diagramming application, a first diagramming application may calculate the transparency property of a shape based upon the shading property of the shape. A second application may not natively support such a calculation and/or relationship. However, an extension provided in the extension section **102** of the self-describing file **100** may define the calculation and/or relationship, thereby providing the second application with the information that may be used to calculate the transparency property to produce a meaningful value. As such, in embodiments, the extension elements of the extension section **102** provide information, such as, but not limited to, formulas or functions that an application may use to properly calculate values for both natively supported and extend features. In doing so, the extension section may thus provide the application with the ability to preserve unknown (e.g., not natively supported) file contents.

In embodiments, the extension element may be an XML element stored in the extension section **102** of the self-describing file **100**. In such embodiments, the extension element may be a top-level element that identifies information (e.g., formulas, functions, additional objects, additional properties,

6

etc.) that may be used by applications accessing the self-describing file **100** to preserve the content of the self-describing file **100**. In such embodiments, the extension element may contain one or more child elements that are used to provide descriptive information for one or more portions of the file content.

In one embodiment, the extension element may contain one or more child elements that identifies a type of section, object, or other type of data stored in the self-describing file **100**. For example, an object element may be a child of the extension element that defines a section of the self-describing file or a type of object stored in the self-describing file **100**. The object element may contain attributes that identify the type of object as well as the type of information that the object stores. In embodiments, the object element may be an XML element. An application may use information from the object element to identify an object stored in the file content.

An example of an object element may be a SectionDef element. In an embodiment where the self-describing file **100** is utilized with a diagramming application, the SectionDef element may identify the following types of objects and/or data: a character or set of characters, a paragraph, a scratch, a connection, a field, a control, a geometry, an action, a layer, a user, a property, a hyperlink, a reviewer, an annotation, an action tag, tabs, etc. Although specific examples are provided, one of skill in the art will appreciate that the SectionDef element is but one type of child element that identifies a specific object or data portion of the self-describing file **100**. Table 1 provides an example embodiment of attributes that a SectionDef element may support.

TABLE 1

Example Embodiment of a SectionDef Element	
Attribute	Description
N	The name of the section.
T	The type of rows that will appear in this section.
S	A style inheritance path to follow. Possible values may be "Line," "Fill," "Text," etc.

In further embodiments, the extension element may also contain one or more child elements that define a property that is stored in the self-describing file **100**. In embodiments, a property element may contain information that an application may use to properly calculate a value for a certain property stored in the self-describing file **100**.

An example of a property element may be a CellDef element. In an embodiment where the self-describing file **100** is utilized with a diagramming application, the CellDef element may define the type of value for a property. Table 2 provides an example embodiment of attributes that a CellDef element may support.

TABLE 2

Example Embodiment of a CellDef Element	
Attribute	Description
N	The name of the CellDef element.
T	The cell type. Possible values of T are: 1. BYTE: 1 byte unsigned byte 2. CHAR: 1 byte signed byte 3. BOOL: 1 byte Boolean 4. PCTU100_5: 1 byte unsigned percent, 0.00 to 100.0, rounded to half-percent, stored 0-255 5. WORD: 2 byte unsigned short 6. SHORT: 2 byte signed short